<table>
<thead>
<tr>
<th>Title</th>
<th>HeyStaks: A Real-World deployment of Social Search</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authors(s)</td>
<td>Smyth, Barry; Coyle, Maurice; Briggs, Peter</td>
</tr>
<tr>
<td>Publication date</td>
<td>2012</td>
</tr>
<tr>
<td>Publication information</td>
<td>RecSys ’12 Proceedings of the sixth ACM conference on Recommender systems</td>
</tr>
<tr>
<td>Conference details</td>
<td>6th ACM Conference on Recommender Systems (RecSys ’12), Dublin, Ireland, 9th-13th September, 2012</td>
</tr>
<tr>
<td>Publisher</td>
<td>ACM</td>
</tr>
<tr>
<td>Item record/more information</td>
<td><a href="http://hdl.handle.net/10197/4353">http://hdl.handle.net/10197/4353</a></td>
</tr>
<tr>
<td>Publisher's statement</td>
<td>This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in RecSys ’12 Proceedings of the sixth ACM conference on Recommender systems <a href="http://doi.acm.org/10.1145/2365952.2366017">http://doi.acm.org/10.1145/2365952.2366017</a></td>
</tr>
<tr>
<td>Publisher's version (DOI)</td>
<td>10.1145/2365952.2366017</td>
</tr>
</tbody>
</table>

Downloaded 2018-12-07T09:07:08Z

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd_oa)

Some rights reserved. For more information, please see the item record link above.
ABSTRACT
The purpose of this paper is to provide a deployment update for the HeyStaks social search system which uses recommendation techniques to add collaboration to mainstream search engines such as Google, Bing, and Yahoo. We describe our results of initial deployments, including an assessment of the quality of HeyStaks’ recommendations, and highlight some lessons learned in the marketplace.

Keywords
social search, deployed application.

1. INTRODUCTION
Recently researchers have begun to recognise the potential of web search as a platform for a more social and collaborative approach to information discovery. For example, researchers in the area of collaborative information retrieval have sought to make web search more collaborative and more social in a variety of ways; see for example [1–3]. The driving insight is that search is an inherently collaborative affair as people frequently search for similar things in similar ways [6, 8, 10, 12]. By embracing collaboration mainstream search deliver improved result relevance especially in the face of ongoing challenges by co called content farms and increasingly aggress SEO (search engine optimization) strategies[1]. For instance, novice searchers have much to gain from the search experiences of more expert searchers. But such experiences and recommendations need to be integrated into mainstream search contexts.

This is a key objective for HeyStaks: an approach to social search that adds a layer of collaboration and recommendation to mainstream search engines like Google and Bing. The recommendation technology that drives HeyStaks has been described in detail elsewhere [5,9–11]. For this purpose

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

RecSys 2010 Dublin, Ireland
Copyright 20XX ACM X-XXXX-XX-X/XX/XX ...$10.00.
Figure 1: The HeyStaks iPad app: (a) the search homescreen; (b) a search results display with recommended communities/staks; (c) A search results page with recommendations from a joined stak; (d) an individual results page.

Figure 3: (a) Query coverage for Google and HeyStaks based on in-app and trending queries; (b) CTR for Google and HeyStaks for All and Top n conditions.
ductise its social search technology and bring it to market described and evaluated elsewhere. these relevance [9] and reputation [4] components have been mended and selected by another. For the interested reader, users when results which originated from one user are recom-

mendations prove to be relevant. The standard bearer for rel-

ence in web search is the click-through rate (CTR) and, all other things being equal, results that attract a higher CTR are likely to be more relevant than those that do not. In this section we calculate two different variations on the CTR measure. First we calculate the CTR across all Google recommendations versus the organic results returned by Google.

3.1 Coverage

By coverage we mean the percentage of search queries for which Google/HeyStaks can return results. Early on coverage was one of the key challenges faced by HeyStaks: in the absence of a critical mass of staks and stak content recommendations were rare and so the potential benefits to new users were limited. This is a classic example of the cold-start problem in conventional recommender systems [7], which we shall return to later.

To evaluate coverage we consider two sources of test queries. First there are the queries submitted by the HeyStaks users themselves – we refer to these as app queries – and they are important because their coverage tells us how often our current users benefit from HeyStaks recommendations. However, there is the concern that these early-adopter users are probably less representative of future users and so it is also important to understand the likely coverage of HeyStaks as it attracts more mainstream users in the future. To evaluate this coverage we use an alternative set of test queries based on the trending query lists available through Google; we refer to these as trending queries.

The results are presented in Figure 3(a). As expected Google offers near perfect coverage for both app and trending queries, although interestingly it fails to hit 100% for the app queries which highlights the somewhat specialised nature of these early-adopter queries. In contrast, as expected HeyStaks does not provide such complete coverage. But it can generate recommendations for more than 50% of app queries and almost 75% of trending queries. In other words, the current users of HeyStaks enjoy at least some recommendations for half of their queries and so the majority of new users will see recommendations during their daily searches. The even higher coverage for the trending queries bodes well for future users and reflects well on some of the bootstrapping techniques that have been used to drive coverage during the early stages of deployment (see Section 4).

3.2 Relevance

Query coverage is important but only if the recommendations proved to be relevant. We refer to this as the All condition. However, this is imperfect because it is clearly biased towards Google’s longer result-lists; for a typical session Google includes 10 results where as HeyStaks includes at most 4 recommendations and usually fewer. As an alternative, we also calculate the CTR for the n HeyStaks recommendations in a session and compare this to the CTR for the top n Google results;
we call this the Top n condition.

The results shown in Figure 3(b) are clearly very positive. In each condition we can see that HeyStaks enjoys a significantly higher CTR than Google. The difference is particularly striking for the Top n Results condition. In this case, HeyStaks enjoys a CTR of more than 80% compared to Google’s 22% CTR on a like-for-like basis. This means that HeyStaks results attract selections almost 4 times as frequently as Google, which is surely a strong indicator that HeyStaks’ more targeted, social recommendations are proving to be more relevant that the one-size-fits-all organic results from leading mainstream search engines.

4. LESSONS LEARNED

In this section we would like to briefly outline some of the lessons that have been learned as a result of our deployment experiences over the past 12 months. Many of these lessons are not unique to the deployment of recommender systems or apps based around recommender systems technologies but we hope that they will nonetheless help others to learn from some of the challenges that we have faced.

Most recommender systems are subject to the cold-start problem and the challenge of making recommendations to users from the start. For HeyStaks this was a considerable challenge since the absence of a critical mass of users and stak content limited recommendation opportunities. To partially address this issue we added a number of high-level recommendation triggers or filters to staks, basically URL patterns that would trigger the promotion of certain types of organic Google or Bing results when present. For example, a Gadgets & Technology might include filters for techcrunch.com, theverge.com, or mashable.com and so ensure that these results would be promoted when returned by Google or Bing. While not strictly speaking social recommendations, it provided an opportunity to kick-start recommendations for early users by including easily sourced and reliable filters in newly created staks.

Another important lesson concerned the need to reduce app complexity. The HeyStaks browser apps contain a wide variety of functions, including allowing users to create and maintain their own staks. The motivation here was obviously to use this as an opportunity to crowd-source the early staks and their content. However, this functionality was rarely used and so increased app complexity. The mobile and tablet apps provided a more streamlined service, removing stak creation features, and instead prioritising the recommendation of pre-existing staks for users to join at search time. As well as simplifying the user experience this also helped ensure that new users were recommended high-quality staks on topics that mattered to them.

Of course this simplification then limited the opportunity to crowd-source new staks which in turn motivated the development of a semi-automatic stak creation workflow for use in-house. The precise details of this are beyond the scope of this short paper but briefly, by harvesting and clustering trending terms from public search and social network sources (Google, Bing, Twitter etc) it was possible to produce a comprehensive list of stak topics. These topics and their associated seed terms were elaborated using, for example, the Bing query suggestion tool to produce a comprehensive set of popular queries. And by submitted these queries to Google’s ISER on a like-for-like basis. This means, etc.) we obtained an arbitrarily large collection of candidate pages. We developed a rating tool to evaluate the quality of these pages, seeding the new staks with those pages that passed the ratings test, and extracting recurring filter urls from these pages. In this way it was possible to create a large volume of staks to cover trending topics and ensure that they were seeded with high quality content and thus capable of generating reliable recommendations from the start.

5. CONCLUSIONS

In this short paper we have provided a brief update on our experiences developing and deploying a recommendation system for social search. Early indications speak to the value of the approach as users benefit from superior search results when compared directly to mainstream search engines like Google. We have also outlined some important deployment lessons that we have learned that may be of interest for other recommender system deployments.

6. REFERENCES